المفيد الضادي في البرمجة بلغة C

تأليف : ماء العينين الطالب اخيار

main.c

```
#include<stdio.h>

main()
{
    instruction;
    ..
    ..
}
```

بعد الترجمة

main.obj

```
0010101000101101
0010111101010010 1
0010001111010100 0
0101101101100101 1
1111100101101001 1
0011101101110101 0
1110101011010101 1
```

بعد الربط

main

الملف التنفيذي

باسم الله الرحمـــــــــــــن الرحيم

يقول العليم الخبير في محكم كتابه العزيز :

{{ اقرأ باسم ربك الذي خلق , خلق الإنسان من علق , اقرأ وربك الأكرم , الذي علم بالقلم علم الإنسان ما لم يعلم }}

الآيات من 1 إلى 5 من سورة العلق

الفصل الأول : عموميات حول لغة السي

معلومات حول الكتاب

- عنوان الكتاب : المعين الضادي في البرمجة بلغة السي
- تاريخ الصدور : 2009.10.15
- المؤلف : ماء العينين طالب الخير
- العمل : طالب بكلية العلوم والتقنيات بمراكش
- الإقامة : المملكة المغربية الشريفة
- رخصة الكتاب : GPL
- للاتصال : shishi-gami@hotmail.fr

فهرس الفصل الأول :

- مقدمة
- لمحة تاريخية لمراحل تطور لغة C
- أهم مميزات لغة C
- لمحة عن محرر النصوص , المفسر و الرابط
- كتابة أول برنامج لك في C
- البنية العامة لبرامج لغة C

مقدمة :

العديد من الإخوة مِنْ مَنْ مَنَّ الله عليهم وعلمهم من علمه قد بلغوا
ولله الحمد مستوى جيدا في البرمجة بكافة ألوانها ومشاربها
وخصوصا في لغة السي، نسأل الله العلي القدير أن يزيدهم من فضله
ويجعلهم القاعدة في المجتمع الإسلامي لا الاستثناء. لكن الملاحظ هو
أن البعض يضنون بعلمهم على إخوانهم ممن لم يتسنى لهم حيازة هته
المعرفة الشيء الذي أدى إلى قلة المراجع العربية الموجودة في هته
اللغة. ولما كانت زكاة علم المرء تعليم غيره، وبناء على طلب بعض
الإخوة ممن تحذوهم رغبة جادة في تعلم البرمجة بلغة السي تم بعون
الله إخراج هذا الكتاب إلى النور تحت رخصة GPL نسأل العلي
القدير التوفيق والسداد والعون والرشاد والنفع للعباد وعلى بركة الله
نبدأ .

لمحة تاريخية لمراحل تطور لغة السي

- ولدت السي سنة 1972 بمختبرات Bell نتيجة أبحاث بريان كيرنغان ودنيس ريتشي (الصورة لدنيس ريتشي مع كين تومبسون)
- صممت أساسا لكتابة ما اعتبر أول نظام التشغيل يونكس حيث كتبه كين تومبسون بادئ الأمر بالأسمبلي ثم أعاد مع دنيس ريتشي كتابة 90 إلى 95 بالمائة من نواته بلغة السي سنة 1973
- مشتقة من BCPL لمارتن ريتشارد و B لكين تومبسون
- بدأ توحيدها سنة 1983 من طرف لجنة X3J11 ل ANSI ( ) American National Standards Institute ليتم إطلاق نسخة السي القياسية سنة 1989والمسماة ANSI-C
- 1988 نشر ريتشي وكيرنغان النسخة الثانية من كتاب « The C Programming Language » والذي اعتبر بمثابة الكتاب المقدس لمبرمجي لغة السي



مميزات لغة السي

- شمولية : فهي غير موجهة نحو مجال معين كما هو الشأن بالنسبة ل Fortran (تطبيقات علمية وتقنية) أو Cobol (تطبيقات تجارية)
- عصرية : فهي مهيكلة, تصريحيه حيث توفر بنيات تحكم و تصاريح مماثلة لنظيراتها ك Fortran, Pascal وAlgol68
- قريبة من الآلة : لقد صممت السي في المقام الأول لبرمجة نظام التشغيل Unix لذا فهي تتوفر على وظائف قريبة من لغة الآلة وتوفر دوال قادرة على الولوج مباشرة وبكل حرية وسهولة ويسر إلى الوظائف الداخلية للحاسوب (إدارة الذاكرة مثلا)

مميزات لغة السي

- سريعة : تستخدم السي مجموعة من الدوال والأوبيرند القريبة من لغة الآلة ما يسمح بكتابة برامج سريعة وفعالة
- مستقلة عن المنصة : رغم قربها من الآلة إلا أن ذلك لا يمنع السي من العمل على أي نظام يمتلك مفسرا لها فقد انطلقت من كونها لغة الأنظمة العاملة ب Unix لتصير اللغة الرسمية في مجال الحواسيب الدقيقة
- محمولة : يمكن استخدام نفس البرنامج المكتوب بالسي في جميع الأنظمة (جميع أنواع العتاد وجميع أنظمة التشغيل) شرط احترام القياس ANSI-C

تنبيــــــه !!!

الحرية المطلقة التي توفرها هته اللغة تفرض توخي الحيطة والحذر فقربها من الآلة سيف ذو حدين ورغم أنها لغة مهيكلة إلى أنها لا تفرض على المبرمج إتباع نمط معين أثناء البرمجة (كالباسكال) فكل شيء مباح وبالتالي فاحتمال ارتكاب أخطاء سيميولوجية و بنيوية كبير جدا ما قد يكلف مدة زمنية ليست بالهينة - تطول أو تقصر حسب طول الكود وتعقيده - في محاولة تعقب الخطأ وتصحيحه ما يستوجب من المبرمج أن يكون على قدر المسؤولية وأن يحرص على أن تكون برامجه خاصة , مميزة , متماسكة ومفهومة .

ملاحظة :

بالنسبة لل Compiler و Texts Editor و Linker فسأورد (بتصرف) تعريفها بإذن الله كما أوردها الأخ خليل أونيس في الطبعة الثانية من كتاب لغة السي الشامل والذي أوصي بشدة بقراءته فلا أبالغ إن قلت أنه أفضل الكتب العربية التي اضطلعت عليها في لغة السي

محرر النصوص : Texts Editor

أول خطوة في البرمجة هي كتابة البرنامج وذلك على أي محرر نصوص مع مراعاة شرطين :

1. أن يتم حفظ مصدر البرنامج بصيغة **c.** مثلا Salam.c
2. أن تكون النصوص مكتوبة بنظام ASCII

**وتجدر الإشارة إلى أنه لا يمكن استخدام محرر Kword في نظام Linux و Word في نظام Windows

Compiler : المفسر/المترجم

تقوم المترجمات بترجمة الملفات المصدرية إلى لغة منخفضة المستوى كلغة التجميع Assembly Language أو إلى لغة الآلة مباشرة ، حيث يتم بعد الترجمة إنشاء ملفات بامتداد Obj. تحتوي على تعليمات التجميع أو الآلة مم يسهل عملية ربط لغتين أو أكثر مع بعضها ، فمثلا يمكن استدعاء دوال من Pascal في لغة ال C

توجد العديد من المترجمات في أغلب الأنظمة، مثلا في أنظمة Windows يوجد المترجم المجاني Dev C++ المقدم من شركة Bloodshed و هناك أيضا Turbo C أحد أقدم المترجمات و الأكثر استعمالا في الجامعات ،غير أن أفضلها المترجم المميز Pelles C

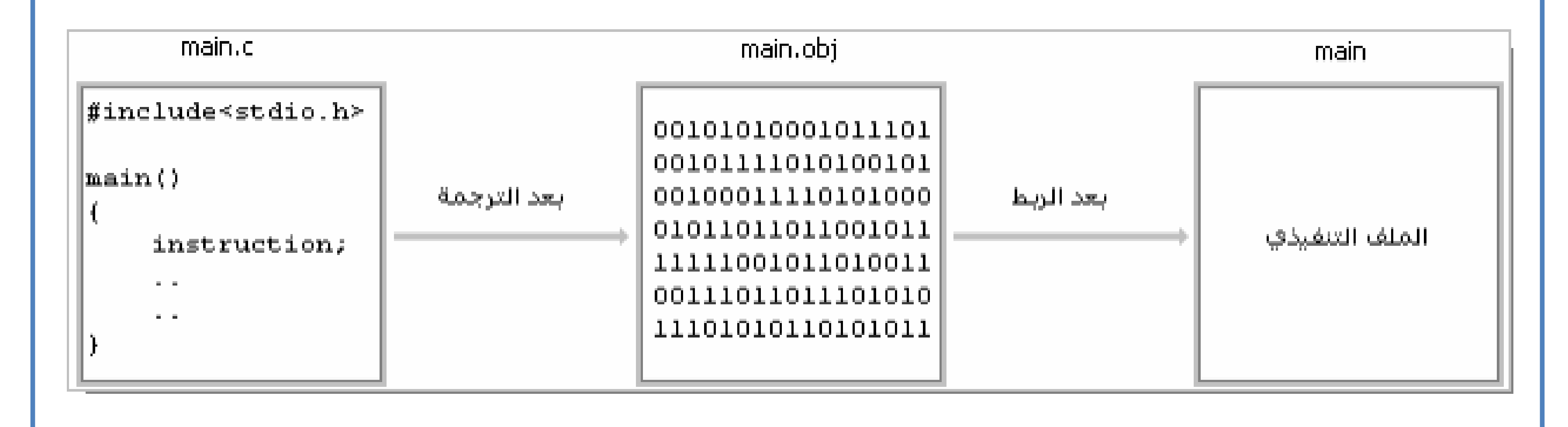
المفسر/المترجم : Compiler

في أنظمة UNIX و Linux لن تحتاج إلى مترجمات لأنها مدمجة مع أي نسخة من نسخ UNIX و Linux، كل ما تحتاجه هو محرر نصوص. وهذا لا يعني أنه لا يوجد مترجمات لتلك الأنظمة، بل يوجد وربما عددها أكثر من التي هي موجودة على نظام Windows.

وتجدر الإشارة إلى أن جميع المترجمات الحديثة تتوفر على IDE أي Integrated Development Environment أي بيئة تطوير متكاملة، تساعد المبرمج في التحرير والترجمة والربط، ففي السابق كانت الترجمة والربط تتم على شكل أوامر أما في IDE فتتم عبر زر واحد من الكيبورد أو نقرة من الماوس، أما كتابة البرامج فتتم عبر محررات نصوص مستقلة عن المترجم، وهذه الظاهرة موجودة إلى الآن في أنظمة UNIX و Linux .

Linker : المربط

يقوم المربط بربط الملفات ذات الامتداد Obj. ثم يعطينا البرامج التنفيذية والتي تكون غالبا بامتداد exe. أو ملفات مكتبات الربط الديناميكية والتي تكون بامتداد dll. ويمكن أن تكون هذه الملفات مكتوبة بمختلف اللغات .



البنية العامة لبرامج السي و كتابة أول برنامج لك فيها

جرت العادة على أن يكون أول برنامج يكتبه المرء هو برنامج الترحيب Hello World وذلك ما سنفعله إن شاء الله مع تعديل طفيف فبدل hello world ستكون كلمة Salam . إذن أول شيء نفعله هو فتح محرر النصوص وكتابة الكود التالي :

```
#include<stdio.h>
                    int main(){
                              printf("Salam");
                    return 0;
                    }
```

نحفظ الكود تحت اسم salam.c ثم بعد ذلك ننفذه بالضغط على زر compile & run إن كان المحرر بالإنجليزية أو compiler et executer إن كان بالفرنسية (هنا العمل يتم بواسطة مفسر مجهز ب IDE)

عند تنفيذ البرنامج سيبدو على الشكل التالي :



يتكون البرنامج في السي من جزأين : جزء خاص بالمعالج ودالة main متبوعة بلامه فاتحة { ومجموعة أوامر وتعليمات تنتهي بلامه مغلقة } . في حالتنا الجملة #include<stdio.h> هي الجزء الخاص بالمعالج ويبدأ البرنامج انطلاقا من السطر int main(){ وينتهي بالامه }

تمثل الجملة printf(”salam”); أمرا للحاسوب بطباعة الكلمة Salam

داخل الدالة main يتم استدعاء مجموعة من الدوال للقيام بعدة وظائف حسب الحاجة كالدالة printf والتي تطبع لنا أي شيء موجود بين علامتي التنصيص ولكي تعمل هته الدوال يجب أن يتعرف عليها المفسر وليتعرف عليها يجب أن نحدد له مصدرها أي المكتبة التي توجد بها وهته المكتبات تكون عادة بامتداد h. كالمكتبة التي قمنا باستعمالها في برنامجنا stdio.h وهي مكتبة شائعة الاستعمال نظرا لأهمية الدوال التي تحتويها وهي دوال الإدخال والإخراج ف std مشتقة من standard و io من in/out .ومن دوال هته المكتبة نجد printf() ،scanf() ،fopen() ،fclose() ،puts() ،gets() ،Fread() ،fwrite()،remove() ...

ولكل دالة من هته الدوال مهمة محددة ومجال تطبيق معين .

ملاحــــظة :

نظرا لأن المترجم الذي أستعمله لا يدعم اللغة العربية وبما أن اللغة الثانية تختلف من بلد لآخر فنجد الإنجليزية عند معظم بلدان المشرق والفرنسية في المغرب العربي ولبنان فقد ارتأيت توفير نسختين من كل برنامج : نسخة إنجليزية وأخرى فرنسية .

البرنامج في حد ذاته لن يتغير، ما سيتغير فقط هو التعاليق والجمل التي ستتم طباعتها. وسأشير إلى لغة البرنامج بتعليق في أقصى اليمين كالتالي :

en// للنسخة الإنجليزية و fr// للنسخة الفرنسية

لاحظوا معي البرنامج التالي :

```
#include<stdio.h>                    //en
Int main(){
Int  age;
Printf("How old are you ? ");
Scanf("%d",&age);
Printf("You are %d years old",age);
system("pause>null");
Return 0;
}
```

```
#include<stdio.h>                    //fr
Int main(){
Int  age;
Printf(”Quel age a vous ? ");
Scanf("%d",&age);
Printf(« Vous avez  %d ans",age);
system("pause>null");
Return 0;
}
```

السطر الأول هو استدعاء للمكتبة stdio.h وذلك لأننا سنستخدم دوال من هته المكتبة والسطر الذي يليه هو بداية البرنامج. إحدى خصائص السي لو تذكرون هي أنها لغة تصريحيه يعني كل دالة سنستخدمها يجب أن نحدد لها مكتبة مرجعية وكذلك كل متغير سنستخدمه يجب التصريح به أولا وهذا ما قمنا به في السطر الثالث حيث عرفنا متغيرا من نوع int أي integer يعني عدد صحيح وأسميناه age .

يقوم عمل البرنامج على سؤال المستخدم عن عمره ثم يؤخذ الكمبيوتر القيمة التي تم إدخالها بواسطة لوحة المفاتيح ومن ثم يطبع الجملة : عمرك هو (العدد الذي ادخله المستخدم) .

إذن في السطر الرابع سنقوم بطباعة الجملة : كم عمرك ؟ وفي السطر الخامس سنقوم بإدخال القيمة التي طبعها المستخدم بالكيبورد وذلك بواسطة الدالة scanf() فالجملة Scanf("%d",&age);

تعني : اقرأ العدد الصحيح الذي تمت طباعته وضع قيمته في المكان المحجوز للمتغير age في الذاكرة.

في السطر السادس استخدمنا الدالة printf() لكن بأسلوب مغاير هته المرة حيث ستطبع ما بين علامتي التنصيص وبدل %d ستضع محتوى المتغير age.

في السطر السابع نجد system("pause>null"); فماذا تعني ؟

من جرب البرنامج الأول سيلاحظ أن البرنامج يظهر ويختفي في لمح البصر فلم ؟ لسبب بسيط هو أن البرنامج أدى وظيفته فأنهى عمله, لكننا لم نر هذا العمل فما السبيل لذلك ؟ الحل يكمن في system("pause"); وهي إحدى دوال النظام لا ينفذ السطر الذي بعدها إلا بعد الضغط على أحد الأزرار أي أننا إذا وضعناها في نهاية البرنامج فلن يغلق هذا الأخير إلا بعد الضغط على زر معين .

لكن ما الفرق بين system("pause"); و system("pause>null"); ؟

جربوا بأنفسكم لتكتشفوا الفرق ☺ .